

バイオグリッド研究会
2004年 11月 27日 (土)

P2P Technologies supporting Grid

首藤 一幸
産業技術総合研究所 グリッド研究センター



Grid
Technology
Research
Center
AIST



内容

- ◆P2Pとは何か

- ◆グリッドに適用されるP2P技術

- ◆P2P分散処理ミドルウェア P3



Personal
Power
Plant

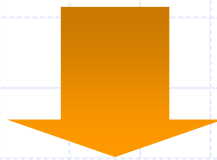


P2P とは何か

What's P2P ?

- ◆ ネットワーク ソフトウェア
- ◆ ファイル交換ソフトウェア？
- ◆ 集中的なサーバが存在しないこと？
- ◆ 非クライアントサーバ型？
- ◆ ...

明確な定義はなく、様々な主張がある。



P2P と呼ばれているシステムを概観して

P2P とは何かを帰納的に考える。



Personal
Power
Plant



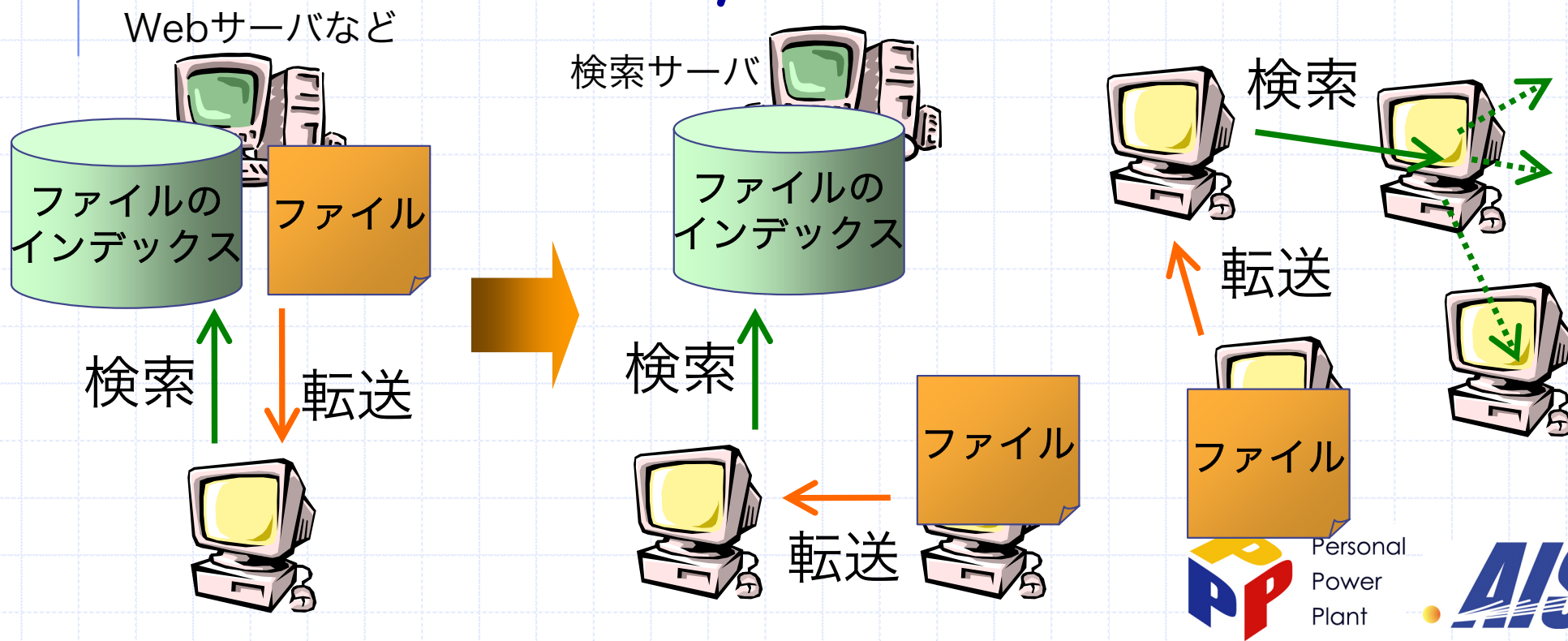
P2P ファイル交換ソフトウェア

- ◆ 各種のファイルを、検索 & 入手、配布するためのソフトウェア
 - 例: Napster, Gnutella, eDonkey, threedegrees, WinMX, Winny, ...
- ◆ 末端（エンドユーザ）の PC 間でファイルを送受信する。

クライアント/サーバ

Hybrid P2P

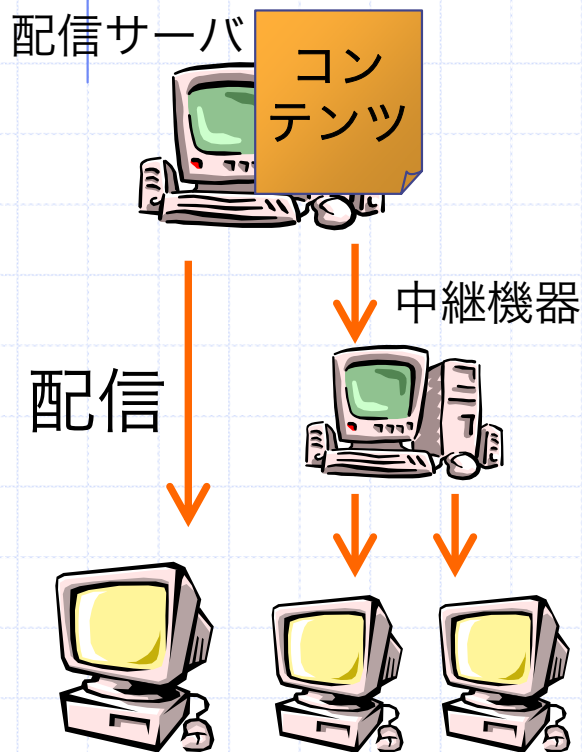
Pure P2P



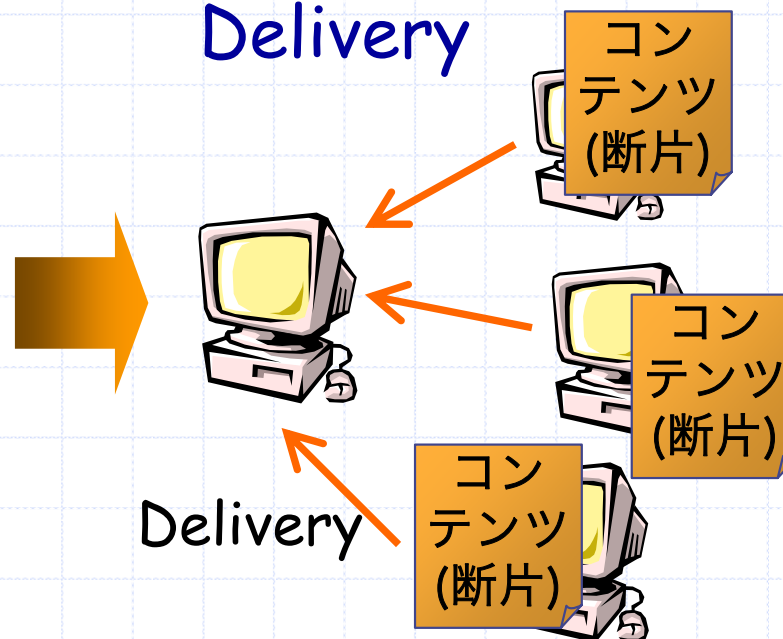
P2P コンテンツ配信, ストリーミング

- ◆ 大容量ファイルや動画を配布/配信する仕掛け。
 - 例: Kontiki社製品, ShareCast, BitTorrent, ...
- ◆ 末端の PC が配信に協力する。

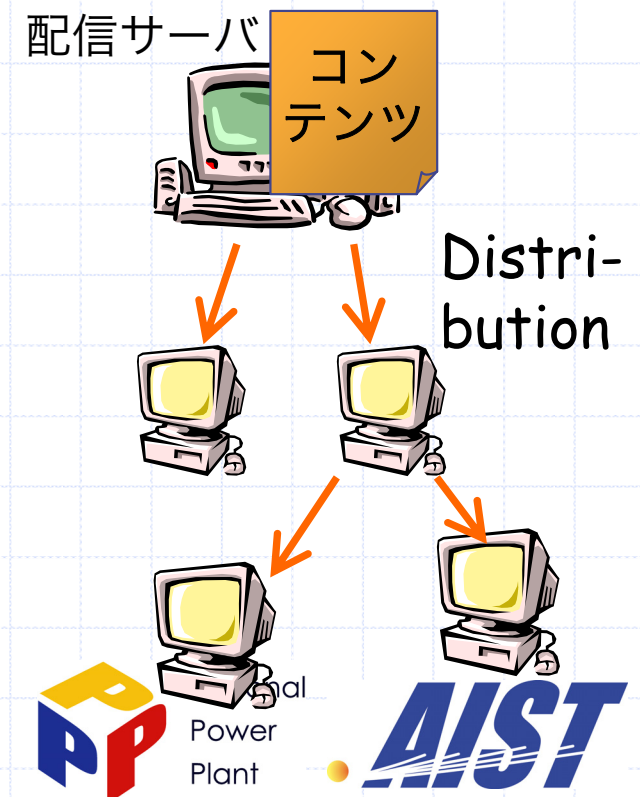
非 P2P



P2P Content Delivery



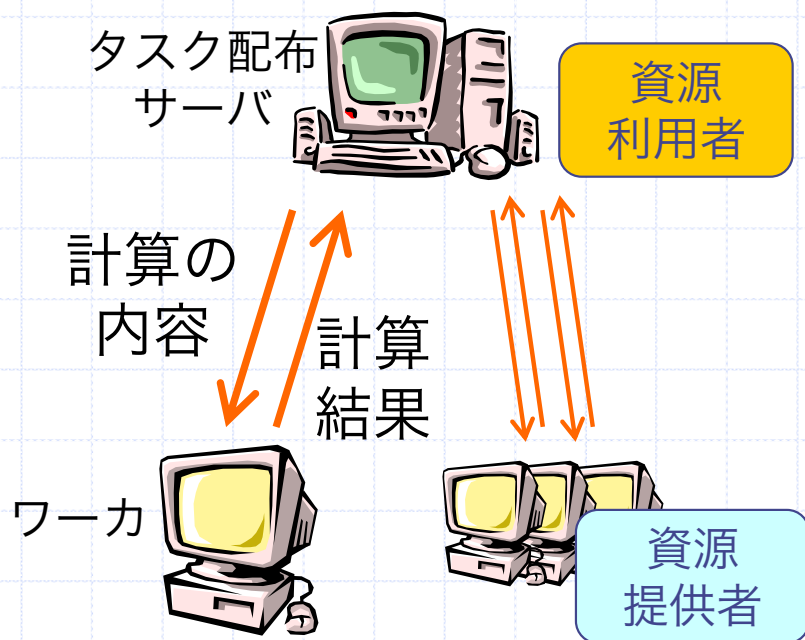
P2P Content Distribution



P2P 分散コンピューティング

- ◆ 手持ちの計算機では足りないような大きな計算を行う。
 - 計算プロジェクトの例: SETI@home, distributed.net
 - ソフトウェアの例: United Devices社 Grid MP, Entropia社 DC Grid, BOINC, XtremWeb, ...
- ◆ 末端の PC が計算やデータ保持を行う。

P2P 分散コンピューティング



P3 (産総研)



Personal
Power
Plant



提供者、利用者の
関係が固定的ではな



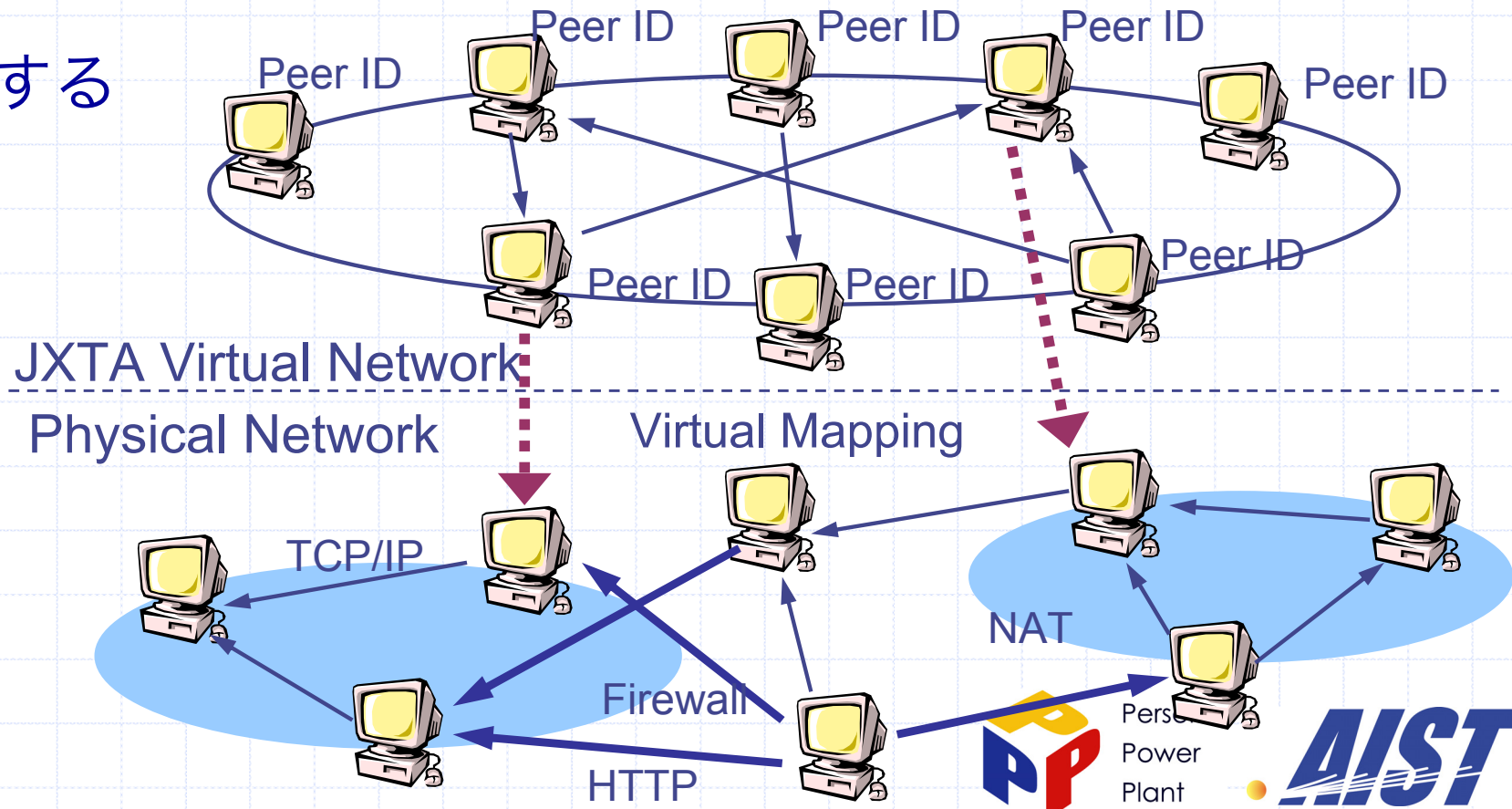
Power
Plant



P2P ミドルウェア

- ◆ P2P ソフトウェアを構築するためのライブラリ、枠組。
- ◆ ピア等の発見, ファイアウォール越しの通信, ピアグループの形成など
 - 例: JXTA, SOBA, Groove Development Kit, Advanced Networking Pack for Windows XP, DECENTRA, ...

JXTAが構成する
Overlay



P2P ○○ いろいろ

◆ P2P グループウェア

- 例: ifreestyle, Ariel AirOne, Groove

◆ P2P 掲示板

- 例: Winny, 新月

◆ P2P Instant Messenger (IM)

- 例: threedegrees

◆ P2P 電話

- 例: Skype

では、P2P とは何か？

- ◆ 集中的なサーバが存在しない。
- ◆ 各ノードの役割が同じ。例えば、あるノードはサーバ/クライアント両方の役割を果たす。
 - IRTF P2P Research Groupの憲章
- ◆ システムの中で固定的なサーバを介さずにピア間で直接通信する場面があるシステム
 - 日経バイト誌 2004年 8月号
- ◆ 末端の機器が重要な役割を果たす分散システム

厳しい

分散コンピューティング
Hybrid P2P
Pure P2P

緩い



Personal
Power
Plant



P2P の性質

◆ スケーラビリティが高い。

- 多くのノードをサポートできる。
 - ◆ 集中的なサーバが存在しない (pure P2P) か、サーバの負荷が軽減される (hybrid P2P) ため。
 - ◆ ときに、数十万ノード以上。

◆ 耐故障性が高い。

- 一部のノードが落ちても、全体は落ちない。

◆ 集中的なサーバの管理コストが不要、または低い。

- P2Pファイル共有と同じサービスを集中サーバで提供すると、数億ドル以上かかるという試算も。

◆ 利用の追跡/管理が困難。

- 集中サーバなしでの、全ノードを対象とした利用履歴の採取やアクセス制限は困難。
- 場合や立場によって、利点であったり難点だったりする。



Personal
Power
Plant



Peer-to-peer から P2P へ

◆“Peer-to-peer”

- 遅くとも 1980年代には使われていた言葉。
 - ◆ cf. UUCP を用いた USENET (1979年)
 - ◆ cf. Macintosh の LocalTalk

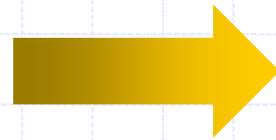
◆“P2P”

- 2000年前後に産み出された言葉。
- ウェブの反動？
 - ◆ ダムクライアントであるウェブブラウザ
- 社会的意義

「P2P」の社会的意義



技術的な性質



社会的な意義

- ◆ 末端ノード間の直接通信
- ◆ 末端ノードが重要な役割を果たす
- ◆ 各ノードの役割が同じ
- ◆ 人と人を直接結びつける技術 (P2P: person-to-person説)
- ◆ 個人中心の社会
- ◆ 階層的ではなくフラットな個人間ネットワーク

「P2P」には、暗に社会的な期待が込められていることが多い。



Personal
Power
Plant



グリッドに適用されるP2P技術

Grid と P2P の関係・相乗作用

- ◆ 極めて直観的
 - 広域分散システム
- ◆ これまであったアクティビティ
 - 一般的、概念的な議論
 - アクティビティ、プロジェクト紹介
 - 要素技術の特性・応用可能性
 - 比較
- ◆ GGF9 workshop (2003年 10月)
 - Peer-to-Peer and Grids: Synergies and Opportunities
 - <http://www-csag.ucsd.edu/P2P-Grid/>
- ◆ APAN and Internet2 joint meeting (2004年 1月)
 - P2P and GRID: Convergence and Challenges
 - <http://apan.net/meetings/honolulu2004/ws-application.htm#appl3>
- ◆ ここでは、Grid/P2P 連係の具体例を概観し、連係パターンを抽出する。



Personal
Power
Plant



Grid / P2P 要素技術

◆ P2P 技術とその性質

- ノードその他資源の **発見** (discovery)
 - ◆ 分散ハッシュテーブル(DHT), ピアのクラスタリング
- **オーバレイ** (overlay)
 - ◆ ピアの論理 ID, NA(P)T 越え, マルチプロトコル
- **スケーラビリティ, 耐故障性, 低管理コスト, ...**

◆ Grid 技術

- 資源の **集約** (aggregation)
- 資源 / 処理の **割り当て**
 - ◆ スケジューリング、ジョブ投入
- **並列 / 分散処理**
- PKI ベースに基づく **認証 / 権限付与**



Personal
Power
Plant



分類

◆ P2P Grid, P2P 分散処理

◆ P2P over Grid パターン

◆ Grid over P2P パターン



Personal
Power
Plant

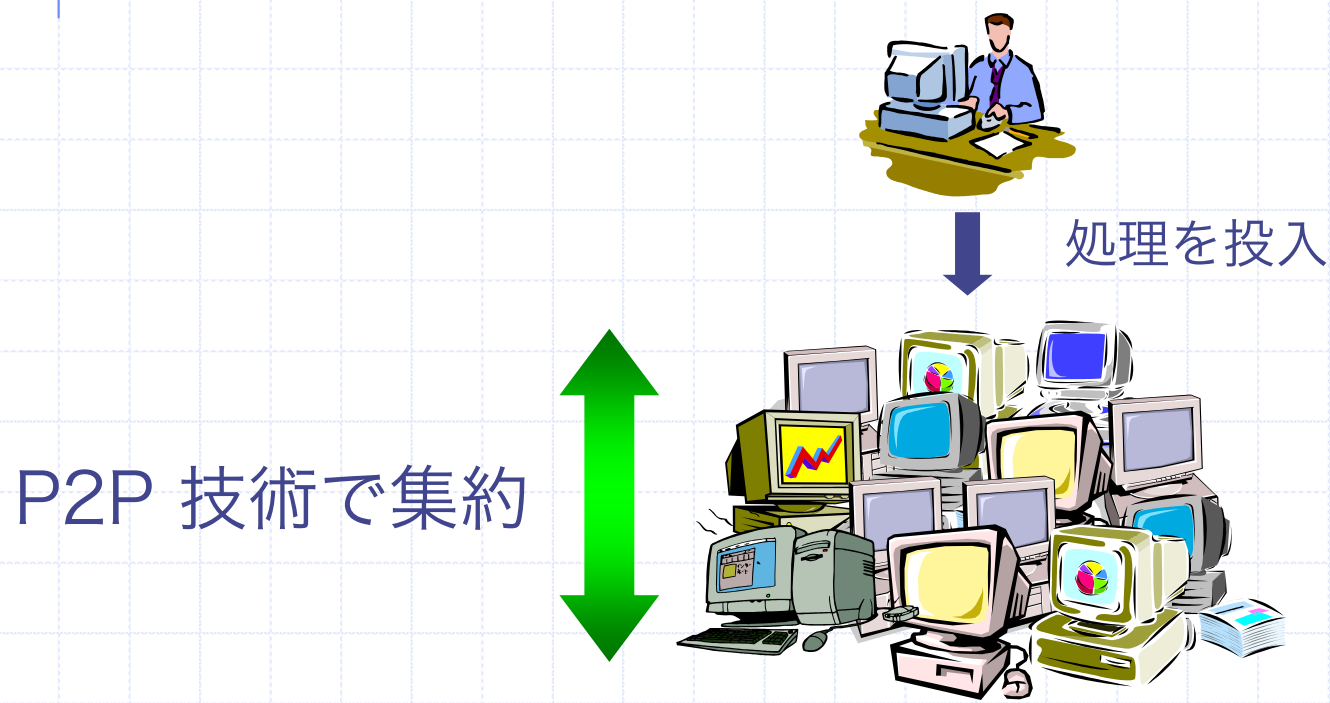


P2P Grid, P2P 分散処理

◆ P2P 技術で計算機資源を集約し、分散処理。

◆ 例

- P2Pミドルウェア JXTA上の分散処理ミドルウェア
 - ◆ P3, JNGI
- (XtremWeb, SETI@home, distributed.net, ...)



Personal
Power
Plant

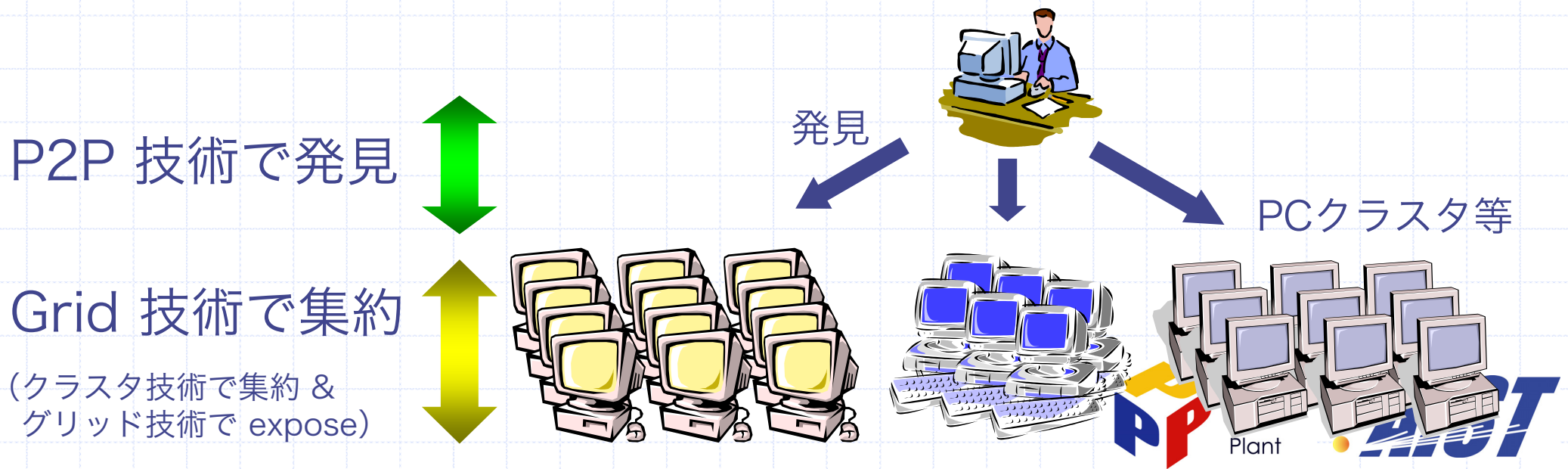


P2P over Grid パターン

◆ Grid 技術で集約した計算機資源を、
P2P 技術で発見する。

◆ 例

- Grid Information Service (例: Globus MDS) に P2P 技術を応用。
 - ◆ P2P GIS (ISTI-CNR, イタリア), GAIS (韓国), 他
- Sun ONE Grid Engine (SGE) + JXTA



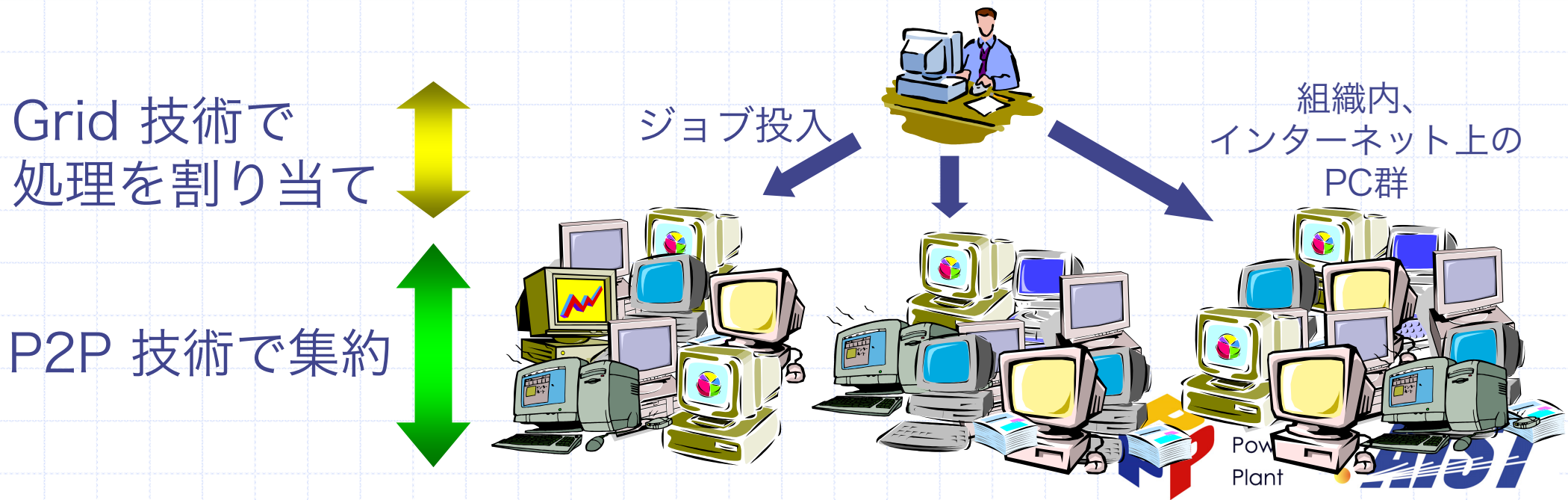
Grid over P2P パターン

◆ P2P 技術で集約した計算機資源に対して、
Grid 技術で処理を割り当てる。

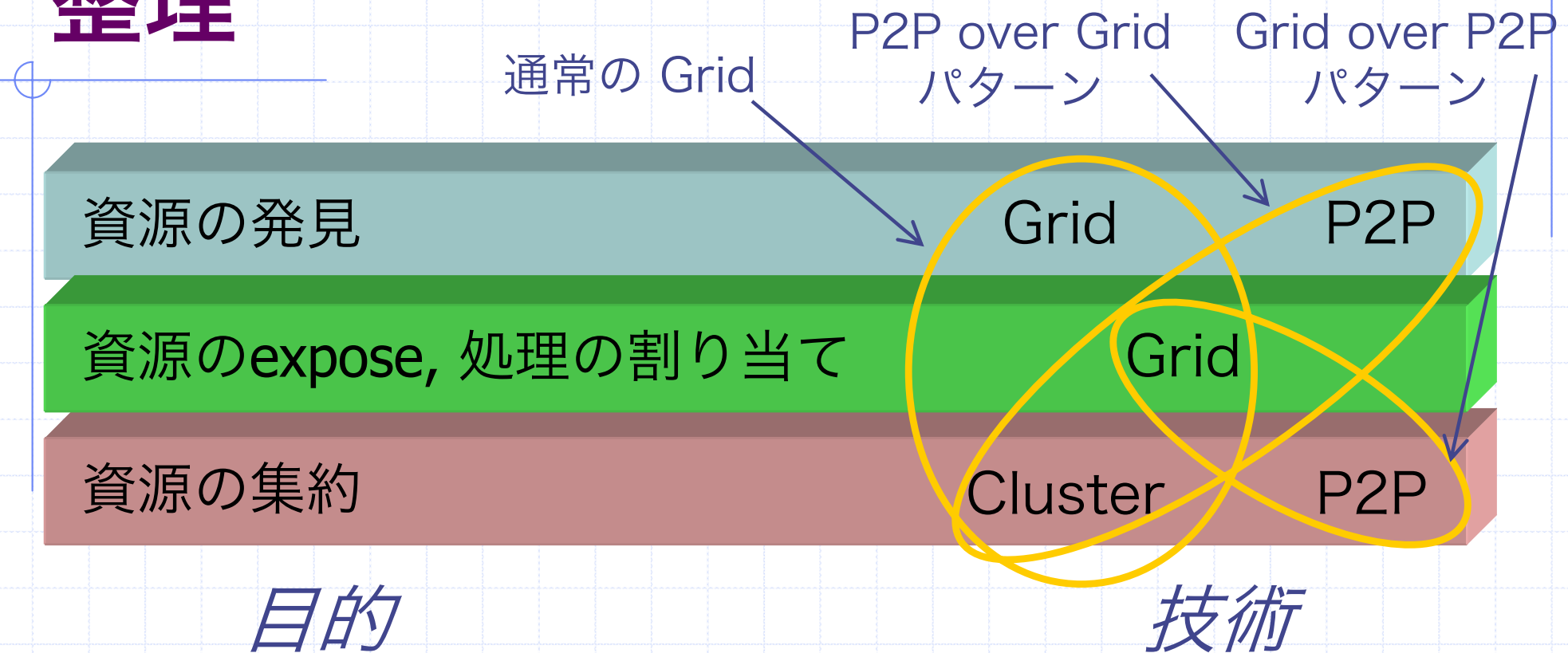
- Grid 技術で、安全に、複数資源にまたがって…等

◆ 例

- United Devices社 Grid MP + Globus Toolkit



整理



- ◆ 他の分類法は？
- ◆ 他の関係パターンは？ 要素技術の適用法は？
- ◆ 新たな関係パターンによる新たな価値？

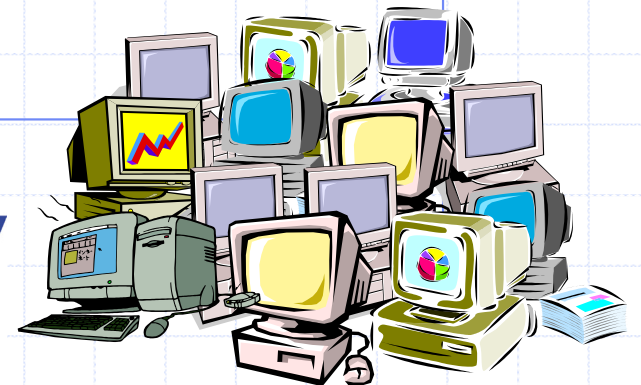
P2P分散処理ミドルウェア P3



Personal
Power
Plant



Grid
Technology
Research
Center
AIST



P3: Personal Power Plant



Personal
Power
Plant

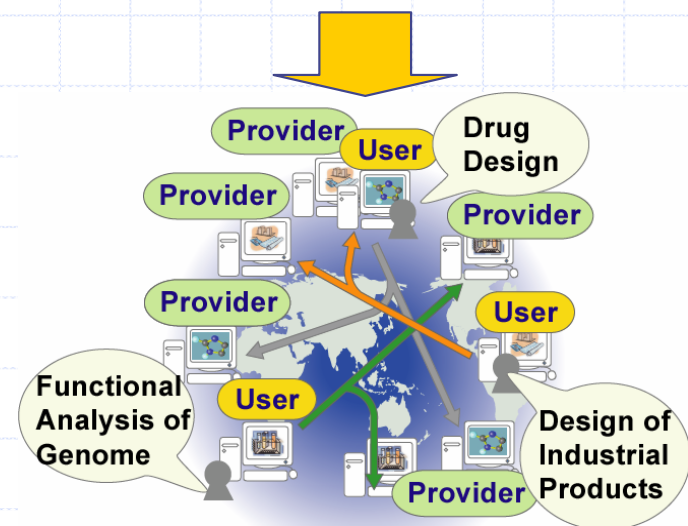
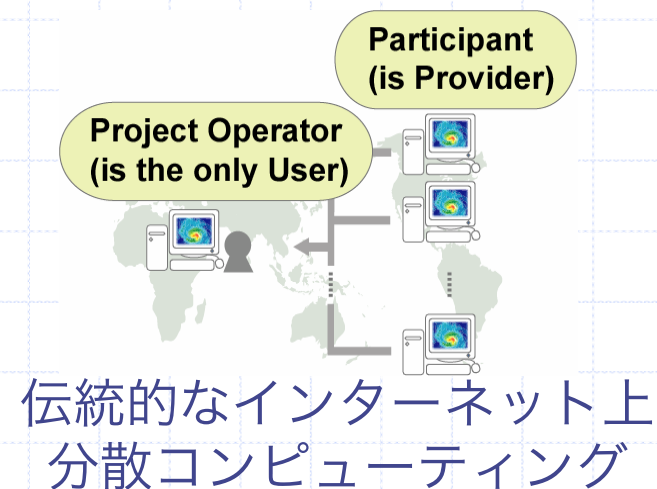
◆ 分散計算ミドルウェア

■ 従来の目標

- ◆ Cycle scavenging
 - 組織内の既存PCの計算能力をかき集める。
- ◆ インターネット上分散コンピューティング
 - 例: distributed.net, SETI@home

■ 将来

- ◆ 多数のPCを集約して、グリッド上の資源として見せる。
 - グリッドミドルウェアとの連携？
cf. Community Scheduler Framework
- ◆ 計算機資源の取引・流通
 - 個人の資源 (C2C, C2B) 、組織の資源 (B2B)
 - 計算能力以外の資源
 - 商取引のための市場と、それを支えるソフトウェアシステム



計算機資源の授受と集約



Personal
Power
Plant



分散処理を支援するP2P技術

- ◆ P3はあらゆる通信に**JXTA**を使用。
- ◆ JXTA: **P2P**ソフトウェアに共通する機能を提供する
プロトコル、参照実装 (Java,C)、**プロジェクト**。
 - 様々なアプリケーション、再利用可能サービスが実装されている。jxta.org上に数十のプロジェクト。
 - ◆ コラボレーション: MyJXTA2, JXCube, ...
 - ◆ 分散処理: JNGI, P3
 - ◆ ファイル共有, ストリーミング, ゲーム, ...
 - 携帯電話機 (J2ME) 向け実装もある。
 - APIが、2001年末の時点ではほぼ固定されている。
- ◆ 語源は“Jxtapose” (並置する)
 - Client-serverといった既存モデルと並存。別に競合するわけではない。
- ◆ 2001年 4月 25日、Sun Microsystems社からアナウンス
- ◆ オープンソースコミュニティ
 - Apache Software License 1.1ほとんどそのままのライセンス条文
 - ...@jxta.orgの登録者は焼く17,000名 (2004年 4月)
 - ダウンロード数 200万以上 (2003年12月)

P3が利用するJXTAの諸機能

- ◆ **アドホック**なピア組織化 (ad-hoc self-organization)
 - 事前設定なしに、発見、通信が可能。
- ◆ **発見** (discovery)
 - PCや投入されたジョブは、PCによって動的に発見される。
- ◆ **ピアグループ** (peer group)
 - ジョブごとにピアグループが作られる。その中で、**コードの配布、ジョブの実行制御、並列処理のためのグループ通信**が行われる。
- ◆ **オーバーレイ** (overlay)
 - ピアには、**下位ネットワークの構造と独立のピアID**が付与される。
cf. IPアドレスやMACアドレス
 - NA(P)T越え、(一方向HTTPの) ファイアウォール越え
 - これによってメッセージパッシング型の並列処理も可能となる。
cf. マスター-ワーカ型



Personal
Power
Plant



JXTAの強み

◆ スケーラビリティ

- 具体的な目標が設定されている：
1,500,000ピア中 300,000ピアがアクティブ
- JXTAが良くなると、その恩恵を受けられる。

◆ 事前の設定が不要

- 他のピアやジョブは、動的に発見できる。
- マルチキャストドメイン（サブネット）を越えるためには、外のピアを指定する必要があるが、ごく容易。マウスクリック程度で済む。

◆ マルチプロトコル

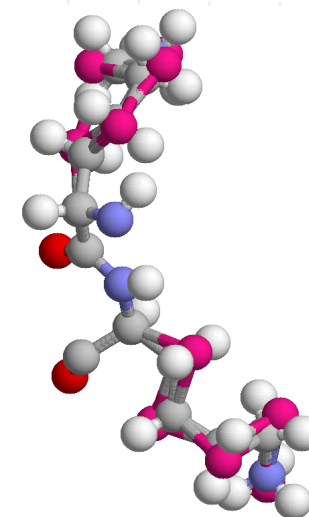
- 下位プロトコルとしてTCP, HTTP, IPマルチキャストをサポート。自動的に使い分け。
- Bluetooth等、他のネットワークも使い得る。



Personal
Power
Plant



要件



求まった
蛋白質の立体構造

◆ アプリケーション中立

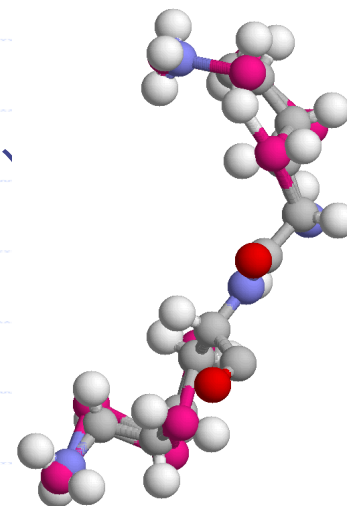
- 従来のインターネット上分散処理プロジェクトでは、少数のアプリケーションがソフトウェアに組み込まれていた。
- P3では、アプリケーションはミドルウェア（P3）とは独立。
 - ◆ アプリケーションは、後からPC群に投入する。

◆ 実用的

- 研究 + 実用ソフトウェア開発
- アプリケーションを載せることで、実用性を確認して、開発にフィードバック。
- 蛋白質の立体構造決定アプリを実装。

◆ スケーラブル

- これまでは、～三十数台で試験。
- スケーラビリティを制限し得る箇所を対象に、試験&改善をしている。
 - ◆ 例: マスタ・ワーカ型並列処理での、マスタの処理効率



要件

◆ PC提供者によるアプリケーションの選択

- 計算機を提供する側で、ジョブを選ぶことができるべき。
- 従来の分散処理プロジェクトでは、選択の余地がなかった。
- P3では、PC提供者は（ジョブの署名を検証した後）その計算を行うか否かを決められる。

◆ インターネット上、組織内の双方への対応

- インターネット上では、悪意あるPC提供者の存在を想定する必要あり。
 - ◆ 計算完了を装う。嘘の結果を返す。サービス不能攻撃をしかける。等。
- P3では、計算結果を検証可能。
 - ◆ 投票方式: 同一の計算を複数のPCに依頼して、結果を照合。
 - ◆ この検証機能は無効化できる。組織内向け。
 - ◆ 検証ロジックを差し替えることも可能。



Personal
Power
Plant



要件

◆ 容易なインストールと自動更新

- PCへのインストール、更新の手間は、台数に比例して増える。
 - ◆ 1台3分だとしても、20台で1時間。100台では…
 - ◆ セキュリティホールが発見されたら…
 - 多くのPCが対応されず、放置されるだろう。
 - 2003年 4月、SETI@homeのクライアントソフトにセキュリティホールが発見された。
 - どのくらいのPCが危険なまま放置されただろうか？
- P3の場合
 - ◆ マウス数クリックでのインストール / 新版への自動更新
 - ◆ 手段: Java Web Start (JWS)



Personal
Power
Plant



P3の設計

◆ 3つのサブシステム

◆ ジョブ管理ソフトウェア

- Host: 計算PCで動作させておくデーモン
- Controller: ジョブを投入するツール

◆ 並列プログラミングライブラリ

- マスタ・ワーカ ライブラリ
- メッセージパッシング ライブラリ
 - ◆ cf. MPI
- エミュレータ
 - ◆ 1台上で並列アプリケーションを動作

◆ ジョブモニタ

- ジョブ、ホストの状態を表示。
- ウェブブラウザで閲覧する。



Personal
Power
Plant



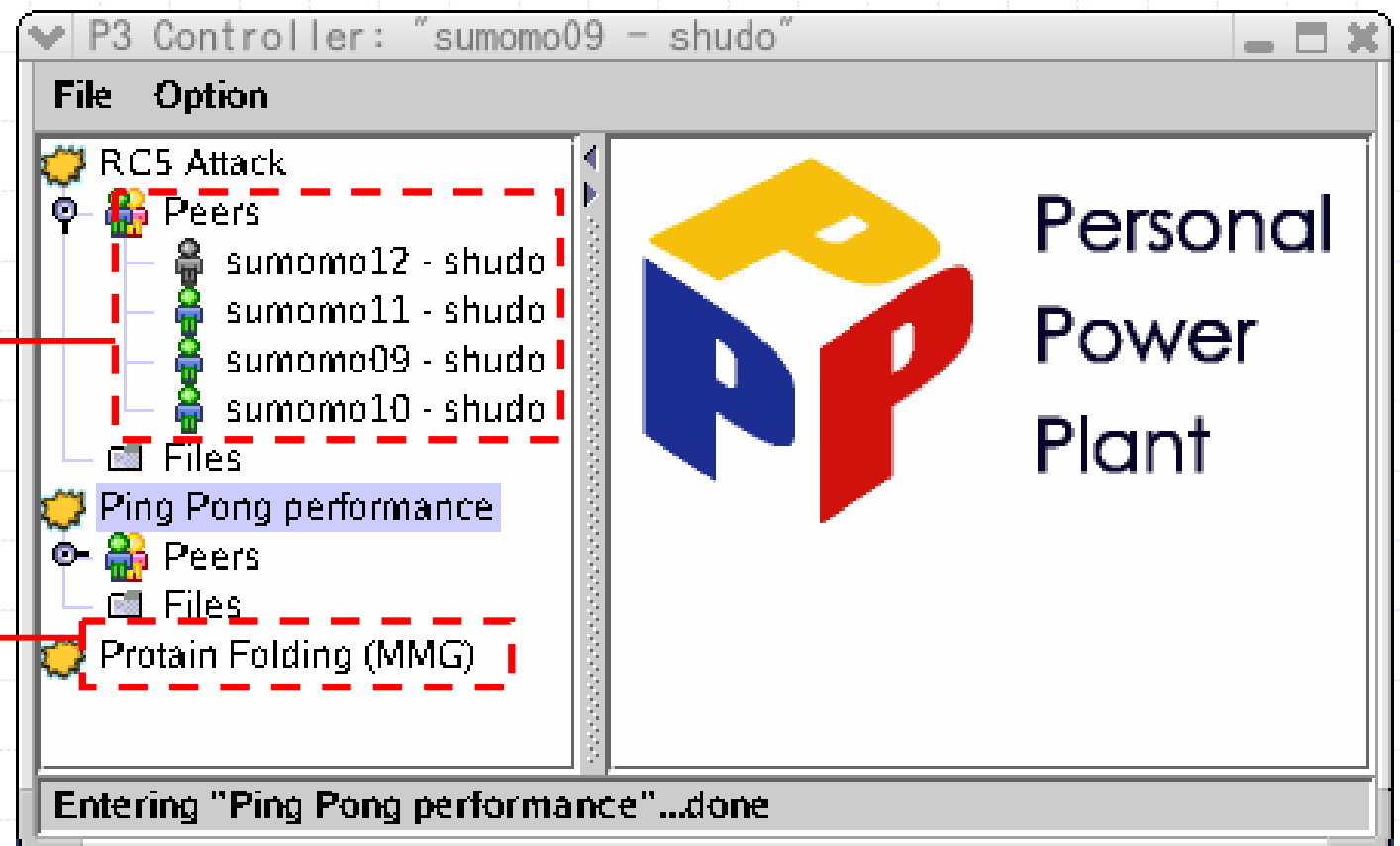
ジョブ管理:

Controller

- ◆ 計算を行いたい利用者が、これを使ってジョブを投入する。

参加しているPC

投入されている
ジョブ

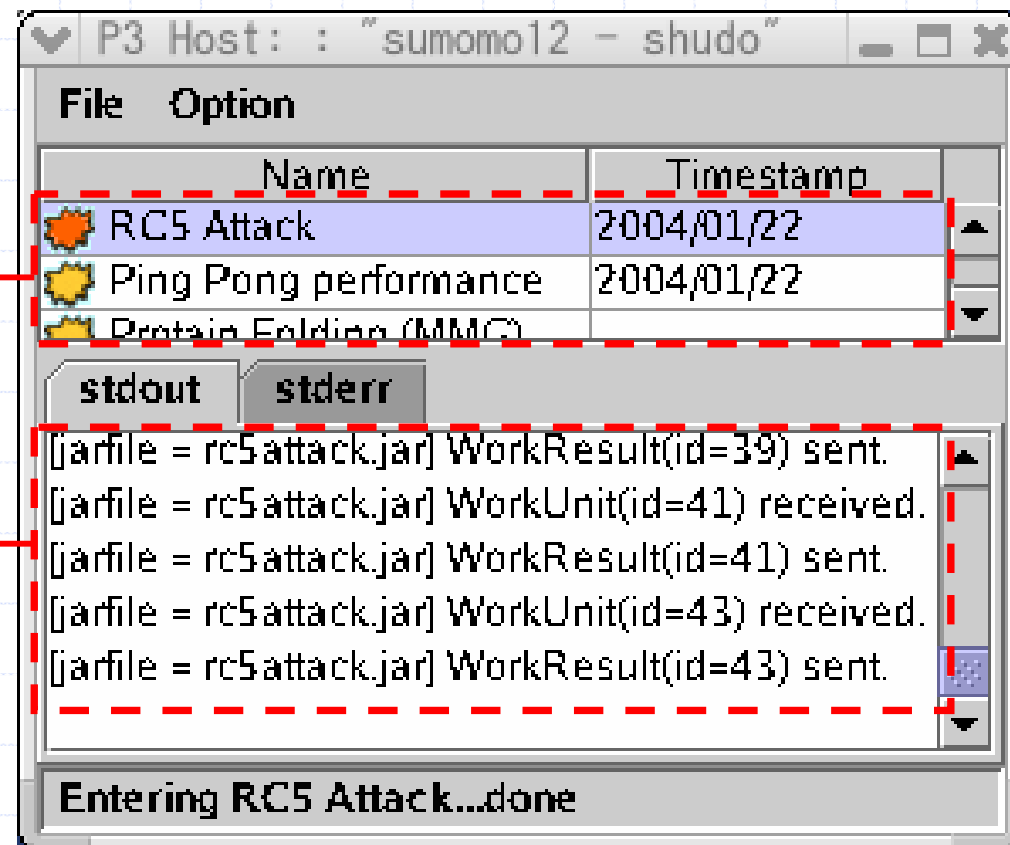


ジョブ管理: Host

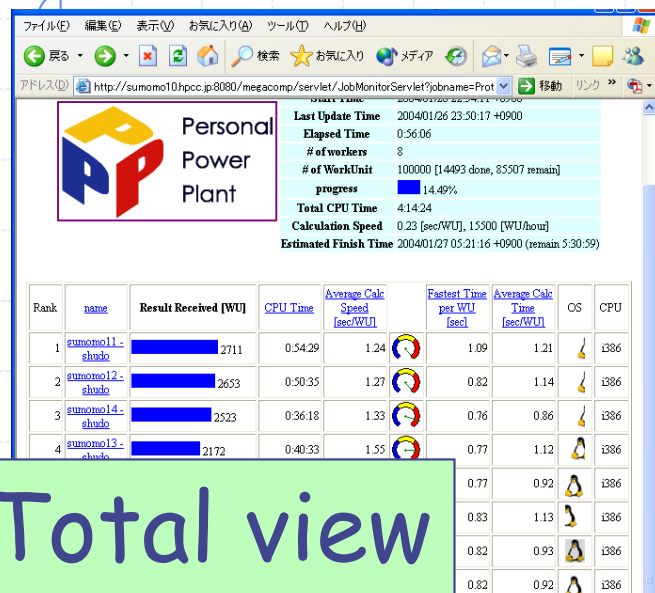
- ◆ 計算を行うPC上で動作させておくデーモン。
 - GUI（ウィンドウ）なしでも動作可能。
その場合Hostは、設定したポリシーに応じて、ジョブへの参加/不参加を決める。
 - 複数のジョブを並行して実行可能。

発見したジョブ

ジョブからの
出力内容

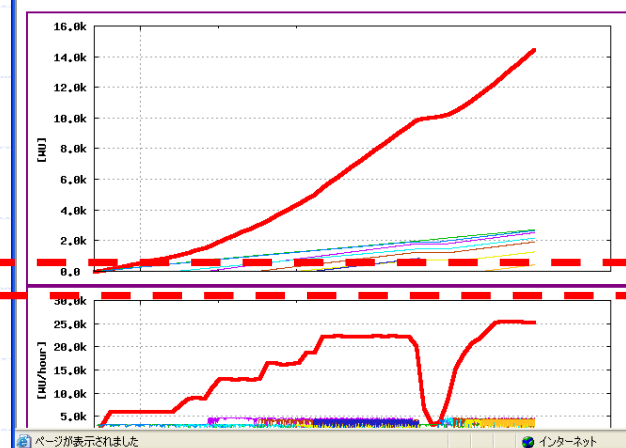
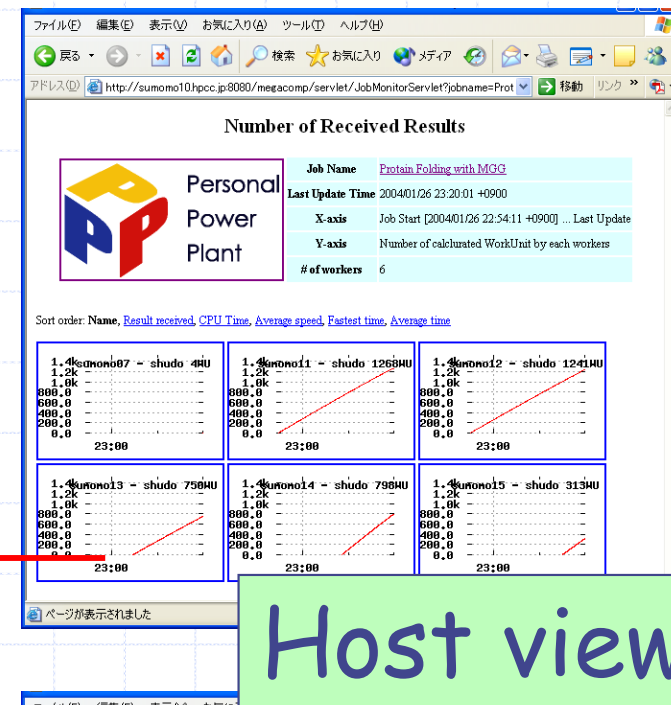


ジョブモニタ



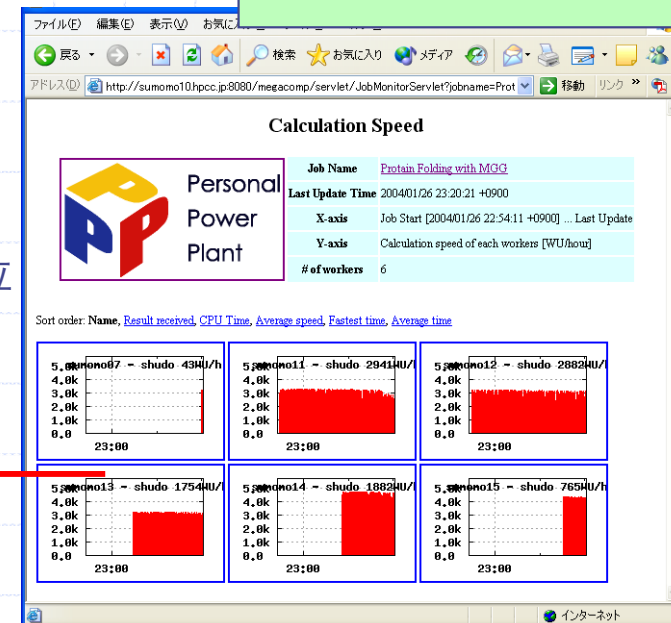
ウェブブラウザ

処理した
ワークユニット数



ワークユニット:
マスタ・ワーカ型
並列処理での
ワーカへの処理配布単位

処理速度



ジョブモニタ



Personal
Power
Plant

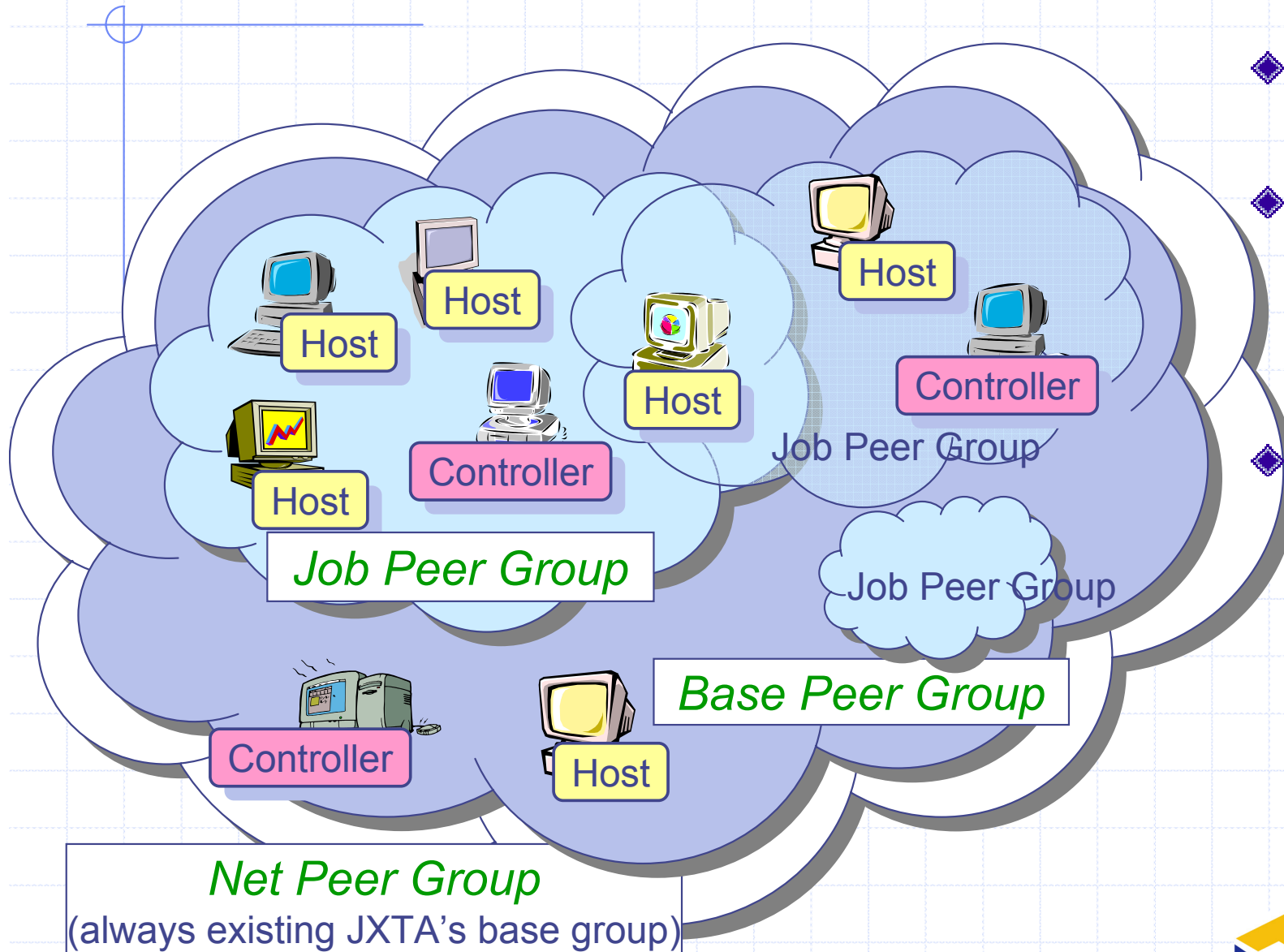
Job Information

Last Update Time	2004/01/27 05:21:16 +0900
Elapsed Time	0:54:29
# of workers	8
# of WorkUnit	100000 [14493 done, 85507 remain]
progress	14.49%
Total CPU Time	4:14:24
Calculation Speed	0.23 [sec/WU], 15500 [WU/hour]
Estimated Finish Time	2004/01/27 05:21:16 +0900 (remain 5:30:59)

Host Information

Rank	name	Result Received [WU]	CPU Time	Average Calc Speed [sec/WU]		Fastest Time per WU [sec]	Average Calc Time [sec/WU]	OS	CPU
1	sumorno11-shudo	2711	0:54:29	1.24		1.09	1.21		i386
2	sumorno12-shudo	2653	0:50:35	1.27		0.82	1.14		i386
3	sumorno14-shudo	2523	0:36:18	1.33		0.76	0.86		i386
4	sumorno13-shudo	2172	0:40:33	1.55		0.77	1.12		i386
5	sumorno15-shudo	1916	0:29:21	1.76		0.77	0.92		i386
6	sumorno07-shudo	1262	0:23:42	2.67		0.83	1.13		i386
7	sumorno06-shudo	837	0:12:57	4.02	down?	0.82	0.93		i386
8	sumorno05-shudo	419	0:06:25	8.03		0.82	0.92		i386

ピアグループの活用方法



◆ Net Peer Group

- JXTAでは常に存在する

◆ Base Peer Group

- P3のピアグループ
- PCやジョブの発見が行われる。

◆ Job Peer Group

- ジョブごとに作られるピアグループ
- ジョブ制御や、並列処理のための通信が行われる。

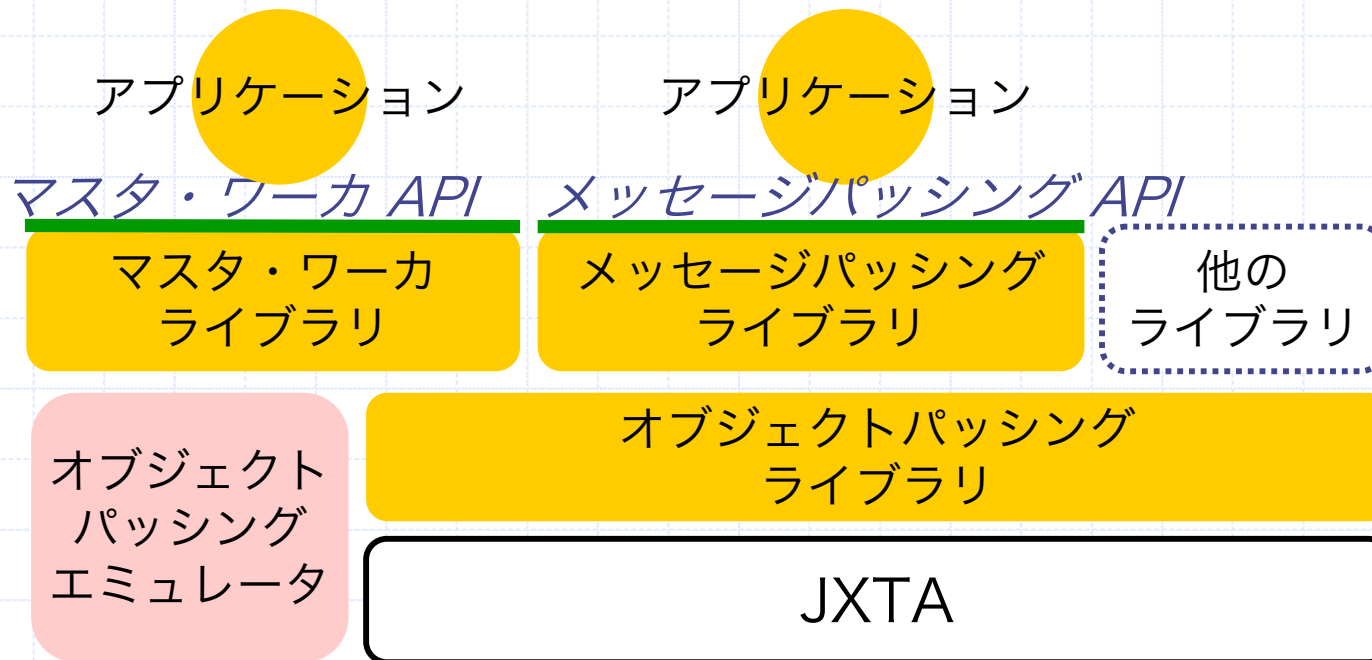
並列プログラミングライブラリ

◆ 主なもの

- マスタ・ワーカ
- メッセージパッシング cf. MPI

◆ エミュレータ

- 1台上で並列アプリを動作させる。
- アプリの開発・デバッグに非常に有用。
 - ◆ マスタ・ワーカライブラリも、エミュレータ上で開発された。



性能測定

◆ JXTAは便利だけれど、遅いんじゃないの？

- 確かに、速くない。けれど、たいてい充分。

◆ 性能測定

- 通信性能
 - ◆ 遅延とスループット
- アプリケーション
 - ◆ RC5暗号化鍵の総当たり探索 cf. distributed.net

◆ 環境

- 2.4 GHz Xeon PC、Gigabit Ethernet
- Linux 2.4.19, Java 2 SDK 1.4.2, JXTA 2.1
- インターネットと比べてネットワークが太い。
しかしそれだけ、P3自体の限界が顕れる。

通信の遅延

◆ 1バイトのピンポン。 片道あたり

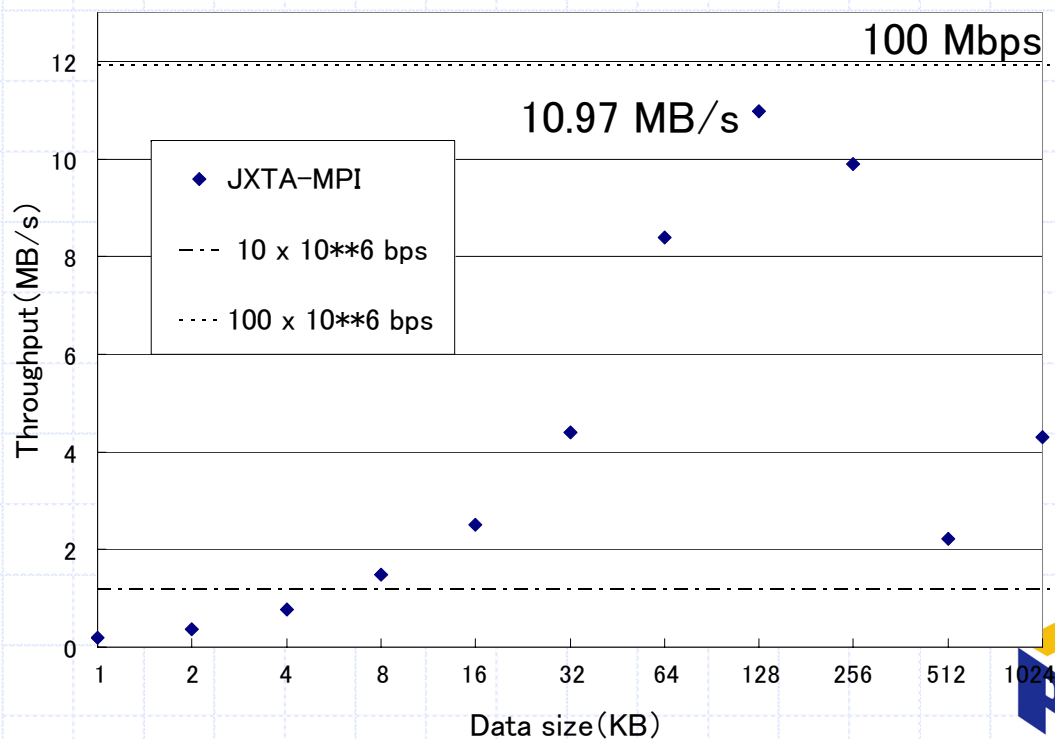
- TCP (in C) 0.062 ミリ秒
- TCP (in Java) 0.064 ミリ秒
- P3のメッセージパッシング 4.5 ミリ秒

◆ 速くない

- マスタ・ワーカ型アプリでは、単位時間あたりのワークユニット処理性能を制限し得る。
- しかし、たいていの用途には充分。
 - ◆ そもそも、ワークユニット1つの処理時間を数秒以上に調整すれば、数ミリ秒は問題とならない。
- そうは言っても、JXTAの改善が望まれる。

通信のスループット

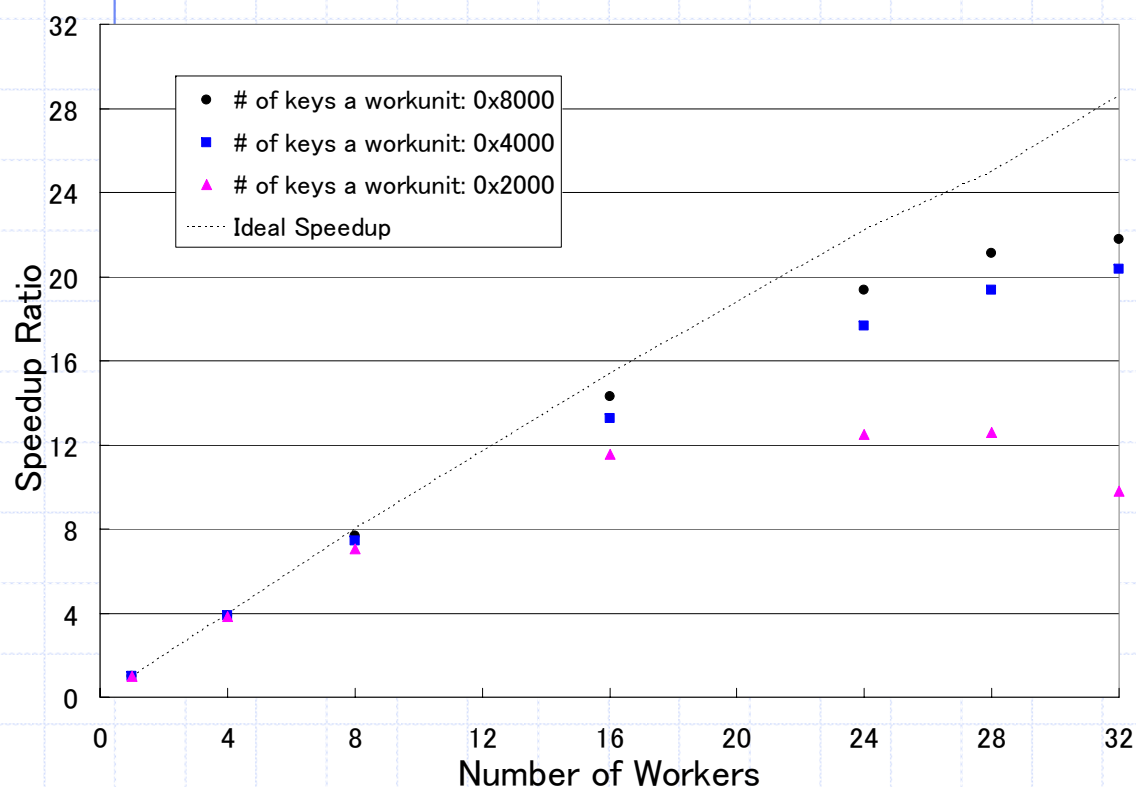
- ◆ メッセージパッシング ライブラリを使用。
- ◆ およそ 100 Mbps (100×10^6 bps)
 - ギガビットイーサネット上には速くないが、この先数年のインターネット接続 (ex. FTTH) を埋めるくらいの性能はある。
- ◆ メッセージサイズが巨大 (512KB~) になると、スループットが低下している。
 - 本来は分割すべき。



アプリケーションでの性能

◆ 小さなワークユニットに対する耐性試験。

- RC5共通鍵暗号方式の暗号化鍵を総当たり探索。
 - ◆ RSA社RC5 Challengeに参加しているdistributed.netと同じ。
- 1.0E1 秒前後の粒度、数十台のPCという条件には充分耐えた。



◆ ワークユニットの粒度:

- 0x8000 : 1.4秒
 - 0x4000 : 0.69秒
 - 0x2000 : 0.36秒
-
- インターネット上分散処理では、通常は数分以上に設定される。

関連アクティビティ

◆ JNGI

- Sun Microsystemsが開発
- JXTAを使用
- ピアグループの使い方が、P3とは異なる。
 - ◆ 計算を行うピアのグループをいくつか事前に作っておく。ピアの効率的な管理が目的？
 - 計算PCはジョブを選べない。
 - cf. P3はジョブごとにピアグループを作り、計算PC (Host) 側でジョブを選べる。

◆ XtremWeb, GreenTea, Javelin, Bayanihan, Ninflet, ...

- それぞれの強みがある。
- 計算PC側でアプリケーションを選べない。
- プログラミングモデルが、マスタ・ワーカや分割統治に限定されている。
- ファイアウォール/NA(P)T越えができない。
 - ◆ Java RMIやTCPをそのまま使うため
- 悪意のあるPCを考慮していない。（か、対応状況不明）

◆ BOINC, United Devices社Grid MP, Entropia社DC Grid

- 略

現状と今後の計画

◆ オープンソース化が完了

- <http://p-three.sf.net/>
- ライセンスは Apache License 2.0
 - ◆ BSDライセンス的で、かなり制約が少ない。
- 2004年 9月公開。

◆ アプリケーションを載せていく

- アプリケーションホスト環境として
(by 阪大 and NICT)

◆ 要望ベースで機能拡張

- 例: ネイティブアプリのホスト



Personal
Power
Plant

